Generative AI workload assessment

AWS Prescriptive Guidance

AWS Prescriptive Guidance: Generative AI workload assessment

Table of Contents

Generative AI workload assessment

Tabby Ward and Deepak Dixit, Amazon Web Services (AWS)

November 2024 (document history)

Generative AI workload assessment is a strategic method aimed at evaluating and improving an organization's preparedness to create or update its generative AI workloads. This assessment is important because incorporating generative AI into business operations can greatly change how things work, and can provide new efficiencies and capabilities. However, to adopt generative AI successfully, it's essential to thoroughly understand current systems and have a clear plan for the future.

Generative AI workloads refer to computational tasks that involve the use of artificial intelligence models that can create new content, such as text, images, code, or other data types. These workloads typically require substantial computing power, specialized hardware such as GPUs, and large datasets for training and inference. Integrating generative AI workloads into operations presents several challenges:

- Infrastructure requirements: Provisioning the significant computational resources and specialized hardware that generative AI models require.
- Data management: Ensuring data quality, privacy, and compliance while handling large datasets.
- Skills gap: Lack of expertise in AI technologies and model deployment.
- Ethical considerations: Addressing bias, fairness, and transparency in AI-generated content.
- Integration complexity: Seamlessly incorporating generative AI into existing workflows and legacy systems.
- Cost management: Balancing the potential benefits with the high costs of implementation and operation.

Overcoming these challenges requires careful planning, investment in infrastructure and talent, and a strategic approach to implementation.

Purpose of this guide

Generative AI is rapidly becoming a critical component across many industries. It provides transformative opportunities but also pose challenges in terms of integration, compliance, and scalability. Many organizations struggle to fully leverage AI due to weak technological foundations, resistance to change, and data quality issues. The generative AI workload assessment addresses these challenges by identifying the requirements for modernization, defining the scope of implementation, and challenging legacy systems and thinking. It also aids in determining minimum viable products (MVPs) and helps you develop a target solution architecture, ensuring a structured and strategic approach to AI adoption.

This guide serves as a structured approach to help organizations navigate the complexities of adopting generative AI technologies. Instead of clearly defining requirements from the outset, the guide assists in:

- Identifying potential use cases for generative AI within your organization.
- Assessing your organization's readiness for generative AI adoption.
- Defining and refining use case goals and stretch goals.
- Determining the scope and requirements for generative AI implementation.
- Developing a target solution architecture.

Target audience and benefits

This assessment is specifically designed for solutions architects, enterprise architects, and application architects who want to evaluate the technical aspects of generative AI workload modernization. It is also valuable for program and people managers who want to gauge their team's overall readiness, resource allocation, and enablement requirements. Industry best practices emphasize the importance of a comprehensive assessment to ensure readiness for AI adoption. This includes evaluating architecture, storage, compliance, integration, testing, deployment, and automation.

Scope

The following topics are in-scope for the generative AI workload assessment method:

- Current generative AI technologies and models (for example, large language models, image generation models)
- Narrow AI applications that use generative techniques
- Integration of generative AI with existing systems and workflows
- Data strategies for training and fine-tuning generative AI models

- Ethical considerations and responsible AI practices for current generative AI applications
- Testing and deployment strategies for generative AI in production environments
- Security and privacy considerations for generative AI implementations
- Performance optimization and scalability of generative AI workloads
- Use cases and applications of generative AI in various industries
- Evaluation of generative AI outputs and quality assurance processes

The following topics are out of scope:

- Artificial general intelligence (AGI) and artificial superintelligence (ASI) scenarios
- Speculative future advancements in AI beyond current generative models
- Quantum computing applications in AI
- Neuromorphic computing and brain-computer interfaces
- Consciousness and self-awareness in AI systems
- Long-term societal impacts of advanced AI beyond current generative AI applications
- Regulatory frameworks for hypothetical future AI technologies
- Philosophical debates on the nature of intelligence and consciousness in machines
- Extreme edge cases or highly speculative use cases of AI
- Detailed technical specifications of proprietary AI models or architectures

Targeted business outcomes

The generative AI workload assessment aims to deliver several targeted outcomes that are crucial for successfully modernizing generative AI workloads. These outcomes ensure that organizations are well prepared to integrate AI technologies effectively and efficiently.

For each targeted outcome, the generative AI workload assessment focuses on:

- Inter-dependencies: Identify and clarify any inter-dependencies between the outcome and other aspects of the modernization process. This includes understanding how one outcome might influence or be influenced by others, to ensure a holistic approach to modernization.
- Stakeholder alignment: Outline strategies to align various stakeholders with each outcome. This involves communicating the value and impact of each outcome to different organizational levels and departments, to foster buy-in and support.
- Prioritization: In cases where multiple use cases or outcomes are identified, provide a framework for prioritizing them based on factors such as business impact, resource requirements, and strategic alignment.
- Continuous improvement: For each outcome, establish mechanisms for ongoing evaluation and refinement. This ensures that the modernization efforts remain adaptive and responsive to changing technological landscapes and business needs.

Here is a detailed discussion of each targeted outcome:

Target architecture

- Definition: The assessment helps define a clear and scalable target architecture for generative AI workloads.
- Components: This includes selecting appropriate cloud services, designing data pipelines, and ensuring system interoperability.
- Benefits: A well-defined architecture supports scalability, reliability, and performance optimization, and provides a strong foundation for modernization.

Customer readiness

- Evaluation: Assess the current state of the organization's infrastructure, processes, and culture to determine readiness for generative AI modernization adoption.

- Criteria: This involves evaluating technical capabilities, data quality, and organizational willingness to embrace change.
- Outcome: Identifying gaps and areas for improvement ensures that the organization is prepared for a smooth transition to modern solution and technologies.

Use case goals and stretch goals

- Use case goals establish clear objectives for target solution implementation, focusing on specific business problems or opportunities.

  A use case goal in the context of generative AI modernization refers to a specific, measurable objective that an organization aims to achieve by implementing generative AI solutions. These goals are typically aligned with broader business objectives and focus on addressing particular challenges or opportunities within the organization. Examples of use case goals might include:

  - Reducing customer service response time by 50 percent by using generative AI-powered chatbots.
  - Improving code review efficiency by 30 percent through generative AI-assisted code analysis.
  - Enhancing fraud detection accuracy by 25 percent by using generative AI pattern recognition.
- Stretch goals define ambitious targets that push the boundaries of what generative AI modernization can achieve within the organization.
- Impact: Setting both achievable and aspirational goals helps align generative AI modernization initiatives with strategic business objectives and encourages innovation.

Effort estimation

- Purpose: Accurate effort estimation aids in resource planning and ensures that projects are delivered on time and within budget.
- Scope: Estimate the resources, time, and budget required to implement the generative AI modernization plan.
- Factors: Consider technical complexity, integration challenges, and potential risks.

Enablement needs

- Training and development: Identify the skills and knowledge required for successful generative AI modernization adoption.

- Resources: Determine the need for training programs, workshops, and other enablement activities.
- Outcome: Ensuring that staff are equipped with the necessary skills enhances the effectiveness of generative AI modernization initiatives and supports long-term success.

Implementation plan

- Roadmap: Develop a detailed plan that outlines the steps required to achieve generative AI modernization.
- Milestones: Define key milestones and deliverables to track progress.
- Benefits: A clear implementation plan provides direction and accountability, and facilitates a structured approach to generative AI modernization.

Assessment considerations and prerequisites

Start with clear use cases

Identify specific business problems or opportunities that generative AI can address. Focus on use cases that align with strategic business goals and offer measurable benefits. Prioritize use cases that target commonly faced challenges within the organization to ensure that the solution architecture can serve as a pattern for multiple scenarios.

Initiating the assessment process with a general understanding of potential generative AI applications is beneficial but not mandatory. The questionnaire that's included with this guide accommodates various levels of preparedness, from organizations that have well-defined use cases to those that have only broad ideas. The assessment process serves to:

- Refine and clarify these initial use case ideas.
- Identify new potential use cases.
- Develop specific, measurable goals for each use case.
- Assess the feasibility and potential impact of each use case.

Let's consider a hypothetical example: A financial services company decides to explore generative AI modernization. They start with a broad idea of improving their customer service and fraud detection processes.

- **Initial assessment**: The questionnaire helps them evaluate their current systems, data quality, and organizational readiness for generative AI adoption.
- **Use case refinement**: Through the assessment process, they refine their initial ideas into two specific use cases:
  - Implementing a generative AI-powered chatbot for customer inquiries
  - Using generative AI for real-time transaction fraud detection
- **Goal setting**: For each use case, they define specific goals:
  - Reduce customer service response time by 40 percent within 6 months
  - Improve fraud detection accuracy by 20 percent and reduce false positives by 15 percent
- **Stretch goals**: They also set these ambitious targets:

  - Achieve 80 percent customer satisfaction with AI-assisted responses
  - Develop a predictive fraud detection model that identifies new fraud patterns
- **MVP definition**: The questionnaire helps them determine an MVP for each use case, focusing on essential features that deliver immediate value.
- **Target architecture**: Finally, they develop a target architecture that supports one or both use cases, and ensures scalability and integration with existing systems.

Ensure business alignment

Align generative AI initiatives with overall business strategy and objectives. For each use case, develop a clear value proposition that demonstrates how generative AI contributes to business growth, efficiency, or innovation. Establish metrics to measure the impact of generative AI implementations on key performance indicators (KPIs).

Implement governance and oversight

Create a cross-functional steering committee to oversee generative AI initiatives. Develop policies and guidelines for responsible AI use, addressing ethical considerations and potential biases. Establish a review process for generative AI projects to ensure compliance with organizational standards and regulatory requirements.

Address data and technical prerequisites

Assess and improve data quality, and implement data governance practices to ensure reliable inputs for generative AI models. Develop a data strategy that addresses data collection, storage, and management that are specific to generative AI needs. Evaluate and enhance data infrastructure to support the volume and velocity of data required for generative AI workloads.

Consider compute resource requirements

Assess current IT infrastructure and identify gaps in computational capacity for generative AI workloads. Plan for scalable compute resources, considering options such as cloud services or on-premises high-performance computing clusters. Optimize resource allocation to balance performance and cost-effectiveness for both training and inference workloads.

Address privacy and security implications

Implement robust security measures to protect sensitive data used in generative AI training and operations. Ensure compliance with data protection regulations such as General Data Protection Regulation (GDPR) or California Consumer Privacy Act (CCPA) when handling personal information. Develop protocols for secure model deployment and monitoring to prevent unauthorized access or misuse of generative AI capabilities.

Engage stakeholders early

Involve key stakeholders from the beginning to gain leadership buy-in and support. Clearly communicate the benefits and potential impact of modernization initiatives, specifically for generative AI workloads. Provide training and resources to help stakeholders understand generative AI technologies and their implications.

Iterate and learn

Adopt an incremental approach that lets you refine target solutions. Use feedback loops to continuously improve workload architecture and processes. Regularly assess the performance and impact of generative AI implementations, and adjust strategies as needed based on real-world results and evolving business needs.

Generative AI workload assessment questionnaire

The following sections provide questions that you can use to evaluate different aspects of generative AI workload modernization for your organization. This comprehensive questionnaire evaluates your organization's readiness to adopt and implement generative AI workloads with questions across key areas, including use cases, architecture, storage, compliance, integration, testing, deployment, and data strategy. By addressing critical aspects of generative AI implementation, from technical infrastructure to regulatory considerations, this questionnaire helps you identify strengths, gaps, and opportunities in your AI modernization journey.

Sections:

You can also download the questionnaire in Microsoft Excel format and use it to record your information.

Download questionnaire

Readiness

Question	Example response
Do you have AWS accounts that can be leveraged for these workloads?	Yes or no.

Question	Example response
Do you have an existing enterprise agreement with AWS?	Yes or no.
How scalable is your current cloud infrastructure to handle generative AI workloads?	Our cloud infrastructure is highly scalable, with automatic scaling capabilities for compute resources and distributed storage systems that are designed to handle large-scale generative AI workloads efficiently.
Do you have data pipeline capabilities for preprocessing and feature engineering at scale?	Our data pipelines use distributed processing frameworks such as Apache Spark for large-scale data preprocessing and feature engineering, with support for both batch and streaming data processing.
Do you have account provisioning and management capability?	Yes or no.
How would you describe your organization's AI literacy and readiness to adopt generative AI technologies?	Our organization has invested heavily in AI education programs, and most technical staff has completed basic AI/ML training. The organization has a culture of innovation that embraces new technologies, including generative AI.
What AI/ML expertise exists within your organization, and how is it distributed?	We have a dedicated AI Center of Excellence with experienced data scientists and ML engineers. We upskill domain experts across different business units to become AI-literate and to identify generative AI use cases.
Do you have a high-level business case that articulates the cloud program objectives, benefits, and cost?	Yes or no.

Question	Example response
What is your time line to take the solution to production?	Weeks, months, and so on.
Has a funding commitment been made by your key stakeholders (for example, CFO, CIT/CTO, COO)?	Yes or no.
How do you ensure compliance with data protection regulations in your generative AI initiatives?	We have a dedicated compliance team that works closely with our AI teams. We conduct regular privacy impact assessments, implement data protection by design principles, and maintain detailed data processing records for all generative AI projects.
How mature are your existing systems that integrate with new generative AI technologies?	Our IT architecture is based on microservices and APIs that allow for flexible integration of new generative AI technologies. These systems are standardized on common data formats and protocols to ensure interoperability.
What experience do you have in operationalizing ML models, and how might this apply to generative AI systems?	We have established MLOps practices, including automated model deployment pipelines, monitoring systems, and A/B testing frameworks. These practices are being adapted to handle the unique requirements of large-scale generative AI models.

Use cases

Question	Example response
What is the primary goal or success criteria of the use case?	To improve customer support response time, increase sales conversions, enhance product recommendations. Also: To improve user

Question	Example response
	satisfaction, task completion rate, response quality, and so on.
How does this use case align with your organization's strategic goals?	This aligns with our strategic goal of enhancing customer satisfaction by reducing response times in customer service.
What is the expected volume of data or requests for the use case?	500 transactions per second (TPS).
What types of data sources are required to support your generative AI workloads?	Internal structured databases (customer records, sales data, and so on); unstructured text data from documents, emails, and social media; audio and video files for speech and image recognition tasks; real-time streaming data from IoT devices and sensors; public datasets and APIs for enrichment.
How frequently do you need to update or refresh data from these sources?	Transactional databases: near real-time updates; document repositories: daily batch updates; social media feeds: hourly updates; IoT sensor data: continuous real-time streaming; public datasets: monthly or quarterly updates.
What data formats do your generative AI models require as input?	Structured data: CSV, JSON, and SQL database tables; text data: plain text, PDF, and HTML; image data: JPEG, PNG, and TIFF; audio data: WAV and MP3; video data: MP4 and AVI.

Question	Example response
What are your key data quality concerns for generative AI workloads?	Completeness: ensuring that no critical fields are missing; accuracy: verifying data correctness and eliminating errors; consisten cy: maintaining uniform formats and values across sources; timeliness: ensuring that data is up to date for real-time inference; relevance : confirming that data aligns with the specific generative AI task.
What are the key performance requireme nts (for example, response time, throughput, accuracy)?	95% accuracy; < 500 ms response time; ability to handle 1000 requests/sec. High accuracy (95%+), moderate accuracy (80-90%), best effort, and so on.
Do you have any other KPIs to measure the success of this use case ?	Key KPIs include error rate reduction, time savings per transaction, and customer satisfact ion scores.
How much model accuracy is desired, and how does it balance with the cost?	High accuracy (>90%) with moderate cost, moderate accuracy (70-80%) with low cost, and so on.
What are the primary use cases or scenarios for the generative AI solution?	Customer service chatbot, content generation, product recommendation, and so on.
What are the target users or personas for the generative AI system?	Customer service agents, marketing team,employees, end users, and so on.
What is the expected volume of requests or users?	1,000 requests per day; 10,000 monthly active users.
Are there any specific use case constraints or requirements?	Real-time response, multi-lingual support, data privacy, and so on.
Do you have an allocated budget for developin g and maintaining the generative AI solution?	The initial development cost is estimated at $200,000, with annual maintenance costs of $50,000.

Question	Example response
What is the projected return on investment (ROI) and payback period for this use case?	Expected ROI of 150% over three years, with a payback period of 18 months.
Are there any hidden costs or potential savings that should be considered?	Potential savings include reduced overtime costs. Hidden costs might involve additional training for staff.
What are the scalability and future expansion possibilities of this generative AI solution?	The solution is designed to scale with our operations, with the possibility of expanding to other departments in the future.
How do you ensure fairness and mitigate bias in your generative AI models?	We plan to mitigate bias through diverse data collection, regular bias audits, and implementation of bias mitigation techniques.
What processes do you have in place for addressing ethical concerns or unintended consequences?	We will manage ethical concerns through an established AI incident response plan, regular ethical risk assessments, an anonymous reporting system for employees, collaboration with external ethics experts, and continuous monitoring and adjustment of deployed models based on feedback.
How do you approach prioritizing and sequencing generative AI workload assessments across different projects and departments in your organization?	By conducting a high-level survey across all departments to identify potential generative AI use cases and evaluating them based on three key criteria: business impact, technical feasibility, and ethical considerations. Projects with high potential impact, lower technical barriers, and minimal ethical concerns are given priority.

Architecture

Question	Example response
What type of generative AI model or architect ure is being considered?	Transformer, convolutional neural network (CNN), recurrent neural network (RNN), decision trees, and so on.
What is the expected scale or volume of data and computations?	Millions of users, petabytes of data, and so on.
What are the hardware requirements (for example, CPUs or GPUs) for training and inference?	High-end GPUs, CPU clusters, cloud instances, and so on.
How will the generative AI model be updated or retrained over time?	Through continuous learning, periodic retraining, manual updates, and so on.
What are the data preprocessing and feature engineering requirements?	Text cleaning, image augmentation, feature selection, and so on.
How will the generative AI system handle edge cases, outliers, or low-confidence inputs?	Through fallback to human oversight, request clarification, and so on.
What are the latency requirements for the generative AI application?	Real-time, near real time, batch processing, and so on.

Storage

Question	Example response
Where will the training data be stored?	In cloud storage (for example, Amazon S3, file storage, block storage, or object storage), in on-premises storage, and so on.

Question	Example response
What are the storage requirements for the training data and model artifacts (for example, capacity, durability, availability)?	Petabyte-scale storage, high durability (99.999999999% durability), high availability, and so on.
What are the data retention and backup requirements for the training data and model artifacts?	Data retention for *x* years, daily backups, off-site backups, and so on.
Which file formats are primarily used for storing your AI training datasets (for example, CSV, JSON, Parquet, HDF5)?	Parquet files for structured data, and HDF5 for large multidimensional arrays and unstructured data such as images and text. We use specialized formats such as TFRecord to optimize data loading during training.
How are your training datasets organized: as individual files, in databases, or using specialized AI data formats?	Small to medium datasets are stored as individual Parquet files in object storage for flexibility. Large datasets are stored in a distributed database (Cassandra) to handle scale.
Do you use any data compression or encoding techniques specifically for generative AI training data?	For tabular data, we use dictionary encoding and bit-packing techniques that are available in Parquet. For images, we use lossy JPEG compression with quality settings optimized for our models.
How do you handle versioning and storage of different iterations of training datasets? What impact does this have on your overall storage needs?	We use a data versioning system (DVC) that is integrated with our ML platform.

Regulations and compliance

Question	Example response
What are the relevant regulations or compliance requirements for the generative AI solution (for example, GDPR, HIPAA, PCI-DSS)?	GDPR for handling personal data, HIPAA for healthcare data, PCI-DSS for payment data, and so on.
What ethical generative AI guidelines or frameworks has your organization adopted?	We implemented our own responsible AI guidelines. All generative AI projects undergo ethical review before approval and deploymen t.
What are the security requirements for the generative AI system?	Data encryption, secure network communica tion, regular security audits.
What are the requirements for data privacy and protection?	Data anonymization, encryption, access control, and so on.
What are the requirements for the solution to handle sensitive or confidential data?	Strict access controls, data masking, data residency requirements, and so on.
How will user authentication and authoriza tion be handled?	By using OAuth, API keys, single sign-on (SSO), and role-based access control (RBAC).
How will the solution be monitored and managed in production?	By using monitoring tools such as Prometheus and Datadog, logging tools such as ELK Stack, alerting systems, and so on.

Integration

Question	Example response
What are the requirements for integrating the generative AI solution with existing systems or data sources?	REST APIs, message queues, database connectors, and so on.

Question	Example response
How will data be ingested and preprocessed for the generative AI solution?	By using batch processing, streaming data, data transformations, and feature engineeri ng.
How will the output of the generative AI solution be consumed or integrated with downstream systems?	Through API endpoints, message queues, database updates, and so on.
Which event-driven integration patterns can be used for the generative AI solution?	Message queues (such as Amazon SQS , Apache Kafka, RabbitMQ), pub/sub systems, webhooks, event streaming platforms.
Which API-based integration approaches can be used to connect the generative AI solution with other systems?	RESTful APIs, GraphQL APIs, SOAP APIs (for legacy systems).
Which microservices architecture component s can be used for the generative AI solution integration?	Service mesh for inter-service communication, API gateways, container orchestration (for example, Kubernetes).
How can hybrid integration be implemented for the generative AI solution?	By combining event-driven patterns for real-time updates, batch processing for historical data, and APIs for external system integration.
How can the generative AI solution output be integrated with downstream systems?	Through API endpoints, message queues, database updates, webhooks, and file exports.
Which security measures should be considered for integrating the generative AI solution?	Authentication mechanisms (such as OAuth or JWT), encryption (in transit and at rest), API rate limiting, and access control lists (ACLs).
How do you plan to integrate open source frameworks such as LlamaIndex or LangChain into your existing data pipeline and generative AI workflow?	We're planning to use LangChain to build complex generative AI applications, particula rly for its agent and memory managemen t capabilities. We aim to have 60% of our generative AI projects using LangChain within the next 6 months.

Question	Example response
How will you ensure compatibility between your chosen open source frameworks and your existing data infrastructure?	We're creating a dedicated integration team to ensure smooth compatibility. By the third quarter, our goal is to have a fully integrate d pipeline that uses LlamaIndex for efficient data indexing and retrieval within our current data lake structure.
How do you plan to leverage the modular components of frameworks such as LangChain for rapid prototyping and experimentation?	We're setting up a sandbox environment where developers can quickly prototype by using LangChain's components.
What is your strategy for keeping up with updates and new features in these rapidly evolving open source frameworks?	We've assigned a team to monitor GitHub repositories and community forums for LangChain and LlamaIndex. We plan to evaluate and integrate major updates quarterly, with a focus on performance improvements and new capabilities.

Testing

Question	Example response
What are the testing requirements (for example, unit testing, integration testing, end-to-end testing)?	Unit testing for individual components, integration testing with external systems, end-to-end testing for critical scenarios, and so on.
How do you ensure data quality and consisten cy across different sources for generative AI training?	We maintain data quality through automated data profiling tools, regular data audits, and a centralized data catalog. We've implemented data governance policies to ensure consistency across sources and to maintain data lineage.
How will the generative AI model be evaluated and validated?	By using a holdout dataset, human evaluation, A/B testing, and so on.

Question	Example response
What are the criteria for evaluating the performance and accuracy of the generative AI model?	Precision, recall, F1 score, perplexity, human evaluation, and so on.
How will edge cases and corner cases be identified and handled?	By using a comprehensive test suite, human evaluation, adversarial testing, and so on.
How will you test for potential biases in the generative AI model?	By using demographic parity analysis, equal opportunity testing, adversarial de-biasing techniques, counterfactual testing, and so on.
Which metrics will be used to measure fairness in the model's outputs?	Disparate impact ratio, equalized odds, demographic parity, individual fairness metrics, and so on.
How will you ensure diverse representation in your test datasets for bias detection?	By using stratified sampling across demographic groups, collaboration with diversity experts, use of synthetic data to fill gaps, and so on.
Which process will be implemented for ongoing monitoring of model fairness post-deployment?	Regular fairness audits, automated bias detection systems, user feedback analysis, periodic retraining with updated datasets, and so on.
How will you address intersectional biases in the generative AI model?	By using intersectional fairness analysis, subgroup testing, collaboration with domain experts on intersectionality, and so on.
How will you test the model's performance across different languages and cultural contexts?	By using multilingual test sets, collaboration with cultural experts, localized fairness metrics, cross-cultural comparison studies, and so on.

Deployment and automation

Question	Example response
What are the requirements for scaling and load balancing?	Intelligent request routing; automatic scaling system; optimizing for fast cold starts by employing techniques such as model caching, lazy loading, and distributed storage systems; designing the system to handle bursty, unpredictable traffic patterns.
What are the requirements for updating and rolling out new versions?	Blue/green deployments, canary releases, rolling updates, and so on.
What are the requirements for disaster recovery and business continuity?	Backup and restore procedures, failover mechanisms, high availability configurations, and so on.
What are the requirements for automating the training, deployment, and management of the generative AI model?	Automated training pipeline, continuous deployment, automatic scaling, and so on.
How will the generative AI model be updated and retrained as new data becomes available?	Through periodic retraining, incremental learning, transfer learning, and so on.
What are the requirements for automating monitoring and management?	Automated alerts, automatic scaling, self-healing, and so on.
What is your preferred deployment environment for generative AI workloads?	A hybrid approach that uses AWS for model training and our on-premises infrastructure for inference to meet data residency requirements.
Are there any specific cloud platforms you prefer for generative AI deployments?	AWS services, particularly Amazon SageMaker AI for model development and deployment, and Amazon Bedrock for foundation models.
What containerization technologies are you considering for generative AI workloads?	We want to standardize on Docker containers that are orchestrated with Kubernetes to

Question	Example response
	ensure portability and scalability across our hybrid environment.
Do you have any preferred tools for CI/CD in your generative AI pipeline?	GitLab for version control and CI/CD pipelines, integrated with Jenkins for automated testing and deployment.
What orchestration tools are you considering for managing generative AI workflows?	Apache Airflow for workflow orchestration, particularly for data preprocessing and model training pipelines.
Do you have any specific requirements for on-premises infrastructure to support generative AI workloads?	We're investing in GPU-accelerated servers and high-speed networking to support on-premises inference workloads.
How do you plan to manage model versioning and deployment across different environments?	We plan to use MLflow for model tracking and versioning, and integrate it with our Kubernetes infrastructure for seamless deployment across environments.
What monitoring and observability tools are you considering for generative AI deployments?	Prometheus for metrics collection and Grafana for visualization, with additional custom logging solutions for model-specific monitoring.
How are you addressing data movement and synchronization in a hybrid deployment model?	We will use AWS DataSync for efficient data transfer between on-premises storage and AWS, with automated synchronization jobs that are scheduled based on our training cycles.
What security measures are you implementing for generative AI deployments across different environments?	We will use IAM for cloud resources, integrated with our on-premises Active Directory to implement end-to-end encryption and network segmentation to secure data flows.

Data strategy

Question	Example response
What specific data types are crucial for your generative AI workloads, and what percentage of these are currently accessible?	Customer call logs and product reviews data are crucial. Currently, 85% of these data types are accessible for our generative AI projects.
How do you ensure and measure the quality of your data?	We have implemented data quality metrics, including completeness, accuracy, consistency, and timeliness. We use automated tools to regularly assess these metrics and have a dedicated team for data cleansing and enrichment.
What percentage of your data meets your quality standards for generative AI use?	Currently, 78% of our data meets our quality standards. We're aiming for 95% within the next 12 months through improved data cleaning processes.
How do you plan to build trust about data usage in generative AI among your stakeholders?	We're implementing an AI ethics board, providing clear explanations of AI decisions, and conducting quarterly AI audits to ensure transparency and fairness.
How comprehensive is your documentation for data sources and lineage?	We maintain a detailed data catalog that includes metadata for all our data sources, including origin, update frequency, and usage. We use data lineage tools to track how data flows and transforms across our systems.
How do you ensure diversity in your datasets to prevent bias in AI models?	We actively source data from diverse demographics and regularly audit our datasets for representational bias. We also use synthetic data generation techniques to balance under-represented categories.

Question	Example response
What is your data refresh rate for critical generative AI models, and how do you determine this frequency?	Critical models are refreshed weekly. This frequency is determined by A/B testing performance metrics, and we aim for no more than 2% degradation between refreshes.
How many versions of critical datasets do you maintain and for how long?	We maintain the last five versions of each critical dataset, with a retention period of 18 months for each version.
How many cross-functional teams are involved in your generative AI initiatives and have access to your data?	We have three cross-functional teams. Each team includes data scientists, domain experts, ethicists, and business analysts.
What data governance policies and practices do you have in place?	We have a cross-functional data governance committee that oversees our data policies. We've implemented role-based access controls, data classification schemes, and regular audits to ensure compliance with our governance framework.
What measures do you have in place to ensure data privacy, obtain proper consent, and maintain confidentiality?	We have implemented a comprehensive data privacy framework aligned with GDPR and CCPA. This includes obtaining explicit consent for data usage, implementing data anonymization techniques, and regular privacy impact assessments.
What percentage of your AI training datasets have been audited for bias in the last quarter?	70% of our AI training datasets were audited for bias last quarter. We're implementing automated bias detection tools to reach 100% quarterly audits.
What is your current data processing capacity, and how much do you project needing for future generative AI workloads?	Our current capacity is 10 TB/day. We project needing 30 TB/day within a year and are scaling our infrastructure to meet this demand.

Question	Example response
What is your strategy for balancing data privacy with the data needs of generative AI models?	We're implementing advanced anonymization techniques and synthetic data generation. Our goal is to increase our usable data for AI by 40% while reducing privacy risks by 60% over the next year.
What percentage of your machine learning (ML) datasets are accurately labeled, and what's your target accuracy rate?	Currently, 85% of our ML datasets are accurately labeled. We're targeting a 95% accuracy rate within the next quarter by employing both human and automated labeling techniques.

Translating assessment insights into actionable outcomes

This section provides a framework for analyzing the questionnaire responses and using those insights to shape the target architecture and other key deliverables of the generative AI modernization initiative. This framework bridges the gap between data collection and implementation, and ensures that the assessment directly informs and drives your modernization strategy.

Target architecture definition:

- Use the questionnaire responses to inform the selection of cloud services and design of data pipelines.
- Make sure that the architecture design supports scalability and interoperability as highlighted in the guide.

Customer readiness evaluation:

- Analyze the questionnaire responses related to current infrastructure, processes, and organizational culture.
- Identify gaps and create a plan to address them. Prioritize gaps that are critical for MVP success.

Use case and stretch goals:

- Extract specific business problems from the questionnaire responses to define clear use case goals.
- Set stretch goals that align with your organization's long-term vision for generative AI modernization.

Effort estimation:

- Use the questionnaire data to estimate resources, time, and budget for both the MVP and full implementation.
- Create a phased approach that starts with the MVP, and outline subsequent phases.

Enablement needs:

- Based on the questionnaire responses, identify skill gaps and training needs.
- Develop a training plan that supports both immediate MVP needs and long-term generative AI adoption.

Implementation plan:

- Create a comprehensive roadmap that starts with the MVP and outlines steps toward full generative AI modernization.
- Define clear milestones and deliverables for each phase of the implementation.

Practical steps:

- **Prioritization matrix**: Create a matrix that maps questionnaire responses to the six outcomes to help prioritize features and efforts.
- **Iterative approach**: Design the MVP to be the first iteration in a series of planned releases, where each release builds toward the full target architecture.
- **Stakeholder alignment**: Use the questionnaire results to align stakeholders on MVP scope and the phased approach to achieving all outcomes.
- **Continuous feedback loop**: Implement mechanisms to gather feedback after MVP deployment, and use insights to refine plans for subsequent phases.
- **Agile implementation**: Adopt an agile methodology that allows for flexibility in addressing all outcomes over time, starting with the most critical outcomes in the MVP.

Next steps

After you complete the generative AI workload assessment, follow these steps:

1. **Deliver a detailed target architecture**
   - Objective: The solution architect creates a comprehensive target architecture that aligns with the organization's goals and the outcomes of the assessment.
   - Components: This architecture includes the design of data ingestion, integration points, and system interoperability to ensure scalability, reliability, and performance optimization.
2. **Explain how specific AWS services fit the use case**
   - Service mapping: Identify and map specific AWS services that best fit the identified use cases.
   - Benefits: Highlight how these services address specific business needs, enhance efficiency, and provide scalability.
3. **Provide optional alternative solutions with pros and cons**
   - Alternatives: Present alternative solutions that could also meet the organization's requirements.
   - Analysis: Offer a detailed analysis of the advantages and disadvantages of each alternative by considering factors such as cost, complexity, and alignment with business goals.
4. **Provide detailed price estimation of AWS services**
   - Cost analysis: Deliver a detailed cost estimation for the proposed AWS services, including potential usage scenarios and pricing models.
   - Budget alignment: Make sure that the cost aligns with the organization's budgetary constraints and provides a clear understanding of the financial implications.
5. **Get feedback on the proposed architecture**
   - Stakeholder engagement: Engage with stakeholders to present the proposed architecture and gather feedback.
   - Iterative improvement: Use the feedback to refine and improve the solution, and confirm that it meets the needs and expectations of all stakeholders.

FAQ

What is the primary objective of the generative AI workload assessment?

The primary objective of the assessment is to evaluate an organization's readiness for modernizing their generative AI workloads, identify use cases, and develop a target solution architecture. It aims to define modernization requirements, determine implementation scope, and prepare for successful generative AI modernization.

Who should use this assessment?

This assessment is for solutions architects, enterprise architects, and application architects who want to assess the technical aspects of generative AI modernization. It is also useful for program managers and people managers to gauge overall readiness, resource allocation, and enablement needs.

What are the key components evaluated in the assessment?

The assessment covers overall readiness, use case, architecture, storage, regulations and compliance, integration, testing, deployment automation, and data strategy. These components are crucial for determining the technical and organizational readiness for generative AI modernization adoption.

How does the assessment help define the target architecture?

The assessment provides a structured approach to evaluate current systems and identify improvements. It helps you select appropriate technologies and design scalable architectures that align with business goals and use case requirements.

What are the benefits of conducting a generative AI workload assessment?

Benefits include enhanced efficiency, improved decision-making, compliance assurance, innovation fostering, and scalability preparation. The assessment establishes a strategic approach to generative AI modernization, and maximizes potential benefits while mitigating risks.

How can organizations ensure successful implementation following the assessment?

Organizations should develop a clear implementation plan that includes defined milestones, engage stakeholders early, and adopt an iterative approach. Establishing a Center of Excellence (CoE) and focusing on talent development are also recommended best practices.

What challenges might organizations face during the assessment?

Challenges might include resistance to change, data quality issues, and compliance complexities. Addressing these challenges requires fostering a culture of innovation, ensuring data readiness, and implementing robust security measures.

How does the assessment address regulatory and compliance requirements?

The assessment evaluates current compliance measures and identifies gaps. It ensures that target solutions adhere to relevant regulations and data privacy laws, and incorporate security best practices to protect sensitive information.

What role does stakeholder engagement play in the assessment process?

Stakeholder engagement is crucial for gaining buy-in, aligning modernization initiatives with business objectives, and ensuring successful implementation. Early involvement and clear communication of benefits are key to overcoming resistance and fostering support.

How can organizations measure the success of their generative AI modernization initiatives after the assessment?

Success can be measured by using key performance indicators (KPIs) that align with business goals. Regular monitoring and evaluation of these metrics help guide decision-making and demonstrate the value of generative AI modernization to stakeholders.

How does the assessment approach differ for organizations of varying sizes (small, medium, or enterprise) or industries?

Small organizations:

- Might have limited resources and expertise for comprehensive assessments
- Likely to focus on specific high-impact use cases instead of enterprise-wide adoption
- Might rely more heavily on third-party tools and services for assessment
- Assessment process might be less formal and more agile

Mid-sized organizations:

- Often have dedicated IT or data teams but might lack specialized AI expertise
- Might take a phased approach, starting with pilot projects in key departments
- Need to balance innovation with existing systems and processes
- Assessment likely involves cross-functional teams

Enterprise organizations:

- Typically have dedicated AI/ML teams and more resources for comprehensive assessment
- Need to consider complex integrations with existing enterprise systems
- Might have industry-specific regulatory requirements to factor in
- Assessment often involves formal governance processes

Resources

- Generative AI on AWS
- AWS offers new artificial intelligence, machine learning, and generative AI guides to plan your AI strategy (AWS blog post)
- Best practices to build generative AI applications on AWS (AWS blog post)
- Generative AI Application Builder on AWS (AWS Solutions Library)
- Generative AI capabilities (*AWS Security Reference Architecture*)
- AWS generative AI best practices framework (*AWS Audit Manager User Guide*)
- Choosing a generative AI service (AWS decision guide)
- What is Amazon Bedrock? (*Amazon Bedrock User Guide*)
- What is Amazon SageMaker AI? (*Amazon SageMaker AI Developer Guide*)

Document history

The following table describes significant changes to this guide. If you want to be notified about future updates, you can subscribe to an RSS feed.

Change	Description	Date
Initial publication	—	November 6, 2024

AWS Prescriptive Guidance glossary

The following are commonly used terms in strategies, guides, and patterns provided by AWS Prescriptive Guidance. To suggest entries, please use the **Provide feedback** link at the end of the glossary.

Numbers

7 Rs

Seven common migration strategies for moving applications to the cloud. These strategies build upon the 5 Rs that Gartner identified in 2011 and consist of the following:

- Refactor/re-architect – Move an application and modify its architecture by taking full advantage of cloud-native features to improve agility, performance, and scalability. This typically involves porting the operating system and database. Example: Migrate your on-premises Oracle database to the Amazon Aurora PostgreSQL-Compatible Edition.
- Replatform (lift and reshape) – Move an application to the cloud, and introduce some level of optimization to take advantage of cloud capabilities. Example: Migrate your on-premises Oracle database to Amazon Relational Database Service (Amazon RDS) for Oracle in the AWS Cloud.
- Repurchase (drop and shop) – Switch to a different product, typically by moving from a traditional license to a SaaS model. Example: Migrate your customer relationship management (CRM) system to Salesforce.com.
- Rehost (lift and shift) – Move an application to the cloud without making any changes to take advantage of cloud capabilities. Example: Migrate your on-premises Oracle database to Oracle on an EC2 instance in the AWS Cloud.
- Relocate (hypervisor-level lift and shift) – Move infrastructure to the cloud without purchasing new hardware, rewriting applications, or modifying your existing operations. You migrate servers from an on-premises platform to a cloud service for the same platform. Example: Migrate a Microsoft Hyper-V application to AWS.
- Retain (revisit) – Keep applications in your source environment. These might include applications that require major refactoring, and you want to postpone that work until a later time, and legacy applications that you want to retain, because there’s no business justification for migrating them.

- Retire – Decommission or remove applications that are no longer needed in your source environment.

A

ABAC

See attribute-based access control.

abstracted services

See managed services.

ACID

See atomicity, consistency, isolation, durability.

active-active migration

A database migration method in which the source and target databases are kept in sync (by using a bidirectional replication tool or dual write operations), and both databases handle transactions from connecting applications during migration. This method supports migration in small, controlled batches instead of requiring a one-time cutover. It's more flexible but requires more work than active-passive migration.

active-passive migration

A database migration method in which the source and target databases are kept in sync, but only the source database handles transactions from connecting applications while data is replicated to the target database. The target database doesn't accept any transactions during migration.

aggregate function

A SQL function that operates on a group of rows and calculates a single return value for the group. Examples of aggregate functions include `SUM` and `MAX`.

AI

See artificial intelligence.

AIOps

See artificial intelligence operations.

anonymization

The process of permanently deleting personal information in a dataset. Anonymization can help protect personal privacy. Anonymized data is no longer considered to be personal data.

anti-pattern

A frequently used solution for a recurring issue where the solution is counter-productive, ineffective, or less effective than an alternative.

application control

A security approach that allows the use of only approved applications in order to help protect a system from malware.

application portfolio

A collection of detailed information about each application used by an organization, including the cost to build and maintain the application, and its business value. This information is key to the portfolio discovery and analysis process and helps identify and prioritize the applications to be migrated, modernized, and optimized.

artificial intelligence (AI)

The field of computer science that is dedicated to using computing technologies to perform cognitive functions that are typically associated with humans, such as learning, solving problems, and recognizing patterns. For more information, see What is Artificial Intelligence?

artificial intelligence operations (AIOps)

The process of using machine learning techniques to solve operational problems, reduce operational incidents and human intervention, and increase service quality. For more information about how AIOps is used in the AWS migration strategy, see the operations integration guide.

asymmetric encryption

An encryption algorithm that uses a pair of keys, a public key for encryption and a private key for decryption. You can share the public key because it isn't used for decryption, but access to the private key should be highly restricted.

atomicity, consistency, isolation, durability (ACID)

A set of software properties that guarantee the data validity and operational reliability of a database, even in the case of errors, power failures, or other problems.

attribute-based access control (ABAC)

The practice of creating fine-grained permissions based on user attributes, such as department, job role, and team name. For more information, see ABAC for AWS in the AWS Identity and Access Management (IAM) documentation.

authoritative data source

A location where you store the primary version of data, which is considered to be the most reliable source of information. You can copy data from the authoritative data source to other locations for the purposes of processing or modifying the data, such as anonymizing, redacting, or pseudonymizing it.

Availability Zone

A distinct location within an AWS Region that is insulated from failures in other Availability Zones and provides inexpensive, low-latency network connectivity to other Availability Zones in the same Region.

AWS Cloud Adoption Framework (AWS CAF)

A framework of guidelines and best practices from AWS to help organizations develop an efficient and effective plan to move successfully to the cloud. AWS CAF organizes guidance into six focus areas called perspectives: business, people, governance, platform, security, and operations. The business, people, and governance perspectives focus on business skills and processes; the platform, security, and operations perspectives focus on technical skills and processes. For example, the people perspective targets stakeholders who handle human resources (HR), staffing functions, and people management. For this perspective, AWS CAF provides guidance for people development, training, and communications to help ready the organization for successful cloud adoption. For more information, see the AWS CAF website and the AWS CAF whitepaper.

AWS Workload Qualification Framework (AWS WQF)

A tool that evaluates database migration workloads, recommends migration strategies, and provides work estimates. AWS WQF is included with AWS Schema Conversion Tool (AWS SCT). It analyzes database schemas and code objects, application code, dependencies, and performance characteristics, and provides assessment reports.

B

bad bot

A bot that is intended to disrupt or cause harm to individuals or organizations.

BCP

See business continuity planning.

behavior graph

A unified, interactive view of resource behavior and interactions over time. You can use a behavior graph with Amazon Detective to examine failed logon attempts, suspicious API calls, and similar actions. For more information, see Data in a behavior graph in the Detective documentation.

big-endian system

A system that stores the most significant byte first. See also endianness.

binary classification

A process that predicts a binary outcome (one of two possible classes). For example, your ML model might need to predict problems such as “Is this email spam or not spam?" or "Is this product a book or a car?"

bloom filter

A probabilistic, memory-efficient data structure that is used to test whether an element is a member of a set.

blue/green deployment

A deployment strategy where you create two separate but identical environments. You run the current application version in one environment (blue) and the new application version in the other environment (green). This strategy helps you quickly roll back with minimal impact.

bot

A software application that runs automated tasks over the internet and simulates human activity or interaction. Some bots are useful or beneficial, such as web crawlers that index information on the internet. Some other bots, known as *bad bots*, are intended to disrupt or cause harm to individuals or organizations.

botnet

Networks of bots that are infected by malware and are under the control of a single party, known as a *bot herder* or *bot operator*. Botnets are the best-known mechanism to scale bots and their impact.

branch

A contained area of a code repository. The first branch created in a repository is the *main branch*. You can create a new branch from an existing branch, and you can then develop features or fix bugs in the new branch. A branch you create to build a feature is commonly referred to as a *feature branch*. When the feature is ready for release, you merge the feature branch back into the main branch. For more information, see About branches (GitHub documentation).

break-glass access

In exceptional circumstances and through an approved process, a quick means for a user to gain access to an AWS account that they don't typically have permissions to access. For more information, see the Implement break-glass procedures indicator in the AWS Well-Architected guidance.

brownfield strategy

The existing infrastructure in your environment. When adopting a brownfield strategy for a system architecture, you design the architecture around the constraints of the current systems and infrastructure. If you are expanding the existing infrastructure, you might blend brownfield and greenfield strategies.

buffer cache

The memory area where the most frequently accessed data is stored.

business capability

What a business does to generate value (for example, sales, customer service, or marketing). Microservices architectures and development decisions can be driven by business capabilities. For more information, see the Organized around business capabilities section of the Running containerized microservices on AWS whitepaper.

business continuity planning (BCP)

A plan that addresses the potential impact of a disruptive event, such as a large-scale migration, on operations and enables a business to resume operations quickly.

C

CAF

See AWS Cloud Adoption Framework.

canary deployment

The slow and incremental release of a version to end users. When you are confident, you deploy the new version and replace the current version in its entirety.

CCoE

See Cloud Center of Excellence.

CDC

See change data capture.

change data capture (CDC)

The process of tracking changes to a data source, such as a database table, and recording metadata about the change. You can use CDC for various purposes, such as auditing or replicating changes in a target system to maintain synchronization.

chaos engineering

Intentionally introducing failures or disruptive events to test a system’s resilience. You can use AWS Fault Injection Service (AWS FIS) to perform experiments that stress your AWS workloads and evaluate their response.

CI/CD

See continuous integration and continuous delivery.

classification

A categorization process that helps generate predictions. ML models for classification problems predict a discrete value. Discrete values are always distinct from one another. For example, a model might need to evaluate whether or not there is a car in an image.

client-side encryption

Encryption of data locally, before the target AWS service receives it.

Cloud Center of Excellence (CCoE)

A multi-disciplinary team that drives cloud adoption efforts across an organization, including developing cloud best practices, mobilizing resources, establishing migration timelines, and leading the organization through large-scale transformations. For more information, see the CCoE posts on the AWS Cloud Enterprise Strategy Blog.

cloud computing

The cloud technology that is typically used for remote data storage and IoT device management. Cloud computing is commonly connected to edge computing technology.

cloud operating model

In an IT organization, the operating model that is used to build, mature, and optimize one or more cloud environments. For more information, see Building your Cloud Operating Model.

cloud stages of adoption

The four phases that organizations typically go through when they migrate to the AWS Cloud:

- Project – Running a few cloud-related projects for proof of concept and learning purposes
- Foundation – Making foundational investments to scale your cloud adoption (e.g., creating a landing zone, defining a CCoE, establishing an operations model)
- Migration – Migrating individual applications
- Re-invention – Optimizing products and services, and innovating in the cloud

These stages were defined by Stephen Orban in the blog post The Journey Toward Cloud-First & the Stages of Adoption on the AWS Cloud Enterprise Strategy blog. For information about how they relate to the AWS migration strategy, see the migration readiness guide.

CMDB

See configuration management database.

code repository

A location where source code and other assets, such as documentation, samples, and scripts, are stored and updated through version control processes. Common cloud repositories include GitHub or Bitbucket Cloud. Each version of the code is called a *branch*. In a microservice structure, each repository is devoted to a single piece of functionality. A single CI/CD pipeline can use multiple repositories.

cold cache

A buffer cache that is empty, not well populated, or contains stale or irrelevant data. This affects performance because the database instance must read from the main memory or disk, which is slower than reading from the buffer cache.

cold data

Data that is rarely accessed and is typically historical. When querying this kind of data, slow queries are typically acceptable. Moving this data to lower-performing and less expensive storage tiers or classes can reduce costs.

computer vision (CV)

A field of AI that uses machine learning to analyze and extract information from visual formats such as digital images and videos. For example, Amazon SageMaker AI provides image processing algorithms for CV.

configuration drift

For a workload, a configuration change from the expected state. It might cause the workload to become noncompliant, and it's typically gradual and unintentional.

configuration management database (CMDB)

A repository that stores and manages information about a database and its IT environment, including both hardware and software components and their configurations. You typically use data from a CMDB in the portfolio discovery and analysis stage of migration.

conformance pack

A collection of AWS Config rules and remediation actions that you can assemble to customize your compliance and security checks. You can deploy a conformance pack as a single entity in an AWS account and Region, or across an organization, by using a YAML template. For more information, see Conformance packs in the AWS Config documentation.

continuous integration and continuous delivery (CI/CD)

The process of automating the source, build, test, staging, and production stages of the software release process. CI/CD is commonly described as a pipeline. CI/CD can help you automate processes, improve productivity, improve code quality, and deliver faster. For more information, see Benefits of continuous delivery. CD can also stand for *continuous deployment*. For more information, see Continuous Delivery vs. Continuous Deployment.

CV

See computer vision.

D

data at rest

Data that is stationary in your network, such as data that is in storage.

data classification

A process for identifying and categorizing the data in your network based on its criticality and sensitivity. It is a critical component of any cybersecurity risk management strategy because it helps you determine the appropriate protection and retention controls for the data. Data classification is a component of the security pillar in the AWS Well-Architected Framework. For more information, see Data classification.

data drift

A meaningful variation between the production data and the data that was used to train an ML model, or a meaningful change in the input data over time. Data drift can reduce the overall quality, accuracy, and fairness in ML model predictions.

data in transit

Data that is actively moving through your network, such as between network resources.

data mesh

An architectural framework that provides distributed, decentralized data ownership with centralized management and governance.

data minimization

The principle of collecting and processing only the data that is strictly necessary. Practicing data minimization in the AWS Cloud can reduce privacy risks, costs, and your analytics carbon footprint.

data perimeter

A set of preventive guardrails in your AWS environment that help make sure that only trusted identities are accessing trusted resources from expected networks. For more information, see Building a data perimeter on AWS.

data preprocessing

To transform raw data into a format that is easily parsed by your ML model. Preprocessing data can mean removing certain columns or rows and addressing missing, inconsistent, or duplicate values.

data provenance

The process of tracking the origin and history of data throughout its lifecycle, such as how the data was generated, transmitted, and stored.

data subject

An individual whose data is being collected and processed.

data warehouse

A data management system that supports business intelligence, such as analytics. Data warehouses commonly contain large amounts of historical data, and they are typically used for queries and analysis.

database definition language (DDL)

Statements or commands for creating or modifying the structure of tables and objects in a database.

database manipulation language (DML)

Statements or commands for modifying (inserting, updating, and deleting) information in a database.

DDL

See database definition language.

deep ensemble

To combine multiple deep learning models for prediction. You can use deep ensembles to obtain a more accurate prediction or for estimating uncertainty in predictions.

deep learning

An ML subfield that uses multiple layers of artificial neural networks to identify mapping between input data and target variables of interest.

defense-in-depth

An information security approach in which a series of security mechanisms and controls are thoughtfully layered throughout a computer network to protect the confidentiality, integrity, and availability of the network and the data within. When you adopt this strategy on AWS, you add multiple controls at different layers of the AWS Organizations structure to help secure resources. For example, a defense-in-depth approach might combine multi-factor authentication, network segmentation, and encryption.

delegated administrator

In AWS Organizations, a compatible service can register an AWS member account to administer the organization's accounts and manage permissions for that service. This account is called the *delegated administrator* for that service. For more information and a list of compatible services, see Services that work with AWS Organizations in the AWS Organizations documentation.

deployment

The process of making an application, new features, or code fixes available in the target environment. Deployment involves implementing changes in a code base and then building and running that code base in the application's environments.

development environment

See environment.

detective control

A security control that is designed to detect, log, and alert after an event has occurred. These controls are a second line of defense, alerting you to security events that bypassed the preventative controls in place. For more information, see Detective controls in *Implementing security controls on AWS*.

development value stream mapping (DVSM)

A process used to identify and prioritize constraints that adversely affect speed and quality in a software development lifecycle. DVSM extends the value stream mapping process originally designed for lean manufacturing practices. It focuses on the steps and teams required to create and move value through the software development process.

digital twin

A virtual representation of a real-world system, such as a building, factory, industrial equipment, or production line. Digital twins support predictive maintenance, remote monitoring, and production optimization.

dimension table

In a star schema, a smaller table that contains data attributes about quantitative data in a fact table. Dimension table attributes are typically text fields or discrete numbers that behave like text. These attributes are commonly used for query constraining, filtering, and result set labeling.

disaster

An event that prevents a workload or system from fulfilling its business objectives in its primary deployed location. These events can be natural disasters, technical failures, or the result of human actions, such as unintentional misconfiguration or a malware attack.

disaster recovery (DR)

The strategy and process you use to minimize downtime and data loss caused by a disaster. For more information, see Disaster Recovery of Workloads on AWS: Recovery in the Cloud in the AWS Well-Architected Framework.

DML

See database manipulation language.

domain-driven design

An approach to developing a complex software system by connecting its components to evolving domains, or core business goals, that each component serves. This concept was introduced by Eric Evans in his book, *Domain-Driven Design: Tackling Complexity in the Heart of Software* (Boston: Addison-Wesley Professional, 2003). For information about how you can use domain-driven design with the strangler fig pattern, see Modernizing legacy Microsoft ASP.NET (ASMX) web services incrementally by using containers and Amazon API Gateway.

DR

See disaster recovery.

drift detection

Tracking deviations from a baselined configuration. For example, you can use AWS CloudFormation to detect drift in system resources, or you can use AWS Control Tower to detect changes in your landing zone that might affect compliance with governance requirements.

DVSM

See development value stream mapping.

E

EDA

See exploratory data analysis.

EDI

See electronic data interchange.

edge computing

The technology that increases the computing power for smart devices at the edges of an IoT network. When compared with cloud computing, edge computing can reduce communication latency and improve response time.

electronic data interchange (EDI)

The automated exchange of business documents between organizations. For more information, see What is Electronic Data Interchange.

encryption

A computing process that transforms plaintext data, which is human-readable, into ciphertext.

encryption key

A cryptographic string of randomized bits that is generated by an encryption algorithm. Keys can vary in length, and each key is designed to be unpredictable and unique.

endianness

The order in which bytes are stored in computer memory. Big-endian systems store the most significant byte first. Little-endian systems store the least significant byte first.

endpoint

See service endpoint.

endpoint service

A service that you can host in a virtual private cloud (VPC) to share with other users. You can create an endpoint service with AWS PrivateLink and grant permissions to other AWS accounts or to AWS Identity and Access Management (IAM) principals. These accounts or principals can connect to your endpoint service privately by creating interface VPC endpoints. For more information, see Create an endpoint service in the Amazon Virtual Private Cloud (Amazon VPC) documentation.

enterprise resource planning (ERP)

A system that automates and manages key business processes (such as accounting, MES, and project management) for an enterprise.

envelope encryption

The process of encrypting an encryption key with another encryption key. For more information, see Envelope encryption in the AWS Key Management Service (AWS KMS) documentation.

environment

An instance of a running application. The following are common types of environments in cloud computing:

- development environment – An instance of a running application that is available only to the core team responsible for maintaining the application. Development environments are used to test changes before promoting them to upper environments. This type of environment is sometimes referred to as a *test environment*.
- lower environments – All development environments for an application, such as those used for initial builds and tests.
- production environment – An instance of a running application that end users can access. In a CI/CD pipeline, the production environment is the last deployment environment.
- upper environments – All environments that can be accessed by users other than the core development team. This can include a production environment, preproduction environments, and environments for user acceptance testing.

epic

In agile methodologies, functional categories that help organize and prioritize your work. Epics provide a high-level description of requirements and implementation tasks. For example, AWS CAF security epics include identity and access management, detective controls, infrastructure security, data protection, and incident response. For more information about epics in the AWS migration strategy, see the program implementation guide.

ERP

See enterprise resource planning.

exploratory data analysis (EDA)

The process of analyzing a dataset to understand its main characteristics. You collect or aggregate data and then perform initial investigations to find patterns, detect anomalies, and check assumptions. EDA is performed by calculating summary statistics and creating data visualizations.

F

fact table

The central table in a star schema. It stores quantitative data about business operations. Typically, a fact table contains two types of columns: those that contain measures and those that contain a foreign key to a dimension table.

fail fast

A philosophy that uses frequent and incremental testing to reduce the development lifecycle. It is a critical part of an agile approach.

fault isolation boundary

In the AWS Cloud, a boundary such as an Availability Zone, AWS Region, control plane, or data plane that limits the effect of a failure and helps improve the resilience of workloads. For more information, see AWS Fault Isolation Boundaries.

feature branch

See branch.

features

The input data that you use to make a prediction. For example, in a manufacturing context, features could be images that are periodically captured from the manufacturing line.

feature importance

How significant a feature is for a model's predictions. This is usually expressed as a numerical score that can be calculated through various techniques, such as Shapley Additive Explanations (SHAP) and integrated gradients. For more information, see Machine learning model interpretability with AWS.

feature transformation

To optimize data for the ML process, including enriching data with additional sources, scaling values, or extracting multiple sets of information from a single data field. This enables the ML model to benefit from the data. For example, if you break down the “2021-05-27 00:15:37” date into “2021”, “May”, “Thu”, and “15”, you can help the learning algorithm learn nuanced patterns associated with different data components.

few-shot prompting

Providing an LLM with a small number of examples that demonstrate the task and desired output before asking it to perform a similar task. This technique is an application of in-context learning, where models learn from examples (*shots*) that are embedded in prompts. Few-shot prompting can be effective for tasks that require specific formatting, reasoning, or domain knowledge. See also zero-shot prompting.

FGAC

See fine-grained access control.

fine-grained access control (FGAC)

The use of multiple conditions to allow or deny an access request.

flash-cut migration

A database migration method that uses continuous data replication through change data capture to migrate data in the shortest time possible, instead of using a phased approach. The objective is to keep downtime to a minimum.

FM

See foundation model.

foundation model (FM)

A large deep-learning neural network that has been training on massive datasets of generalized and unlabeled data. FMs are capable of performing a wide variety of general tasks, such as understanding language, generating text and images, and conversing in natural language. For more information, see What are Foundation Models.

G

generative AI

A subset of AI models that have been trained on large amounts of data and that can use a simple text prompt to create new content and artifacts, such as images, videos, text, and audio. For more information, see What is Generative AI.

geo blocking

See geographic restrictions.

geographic restrictions (geo blocking)

In Amazon CloudFront, an option to prevent users in specific countries from accessing content distributions. You can use an allow list or block list to specify approved and banned countries. For more information, see Restricting the geographic distribution of your content in the CloudFront documentation.

Gitflow workflow

An approach in which lower and upper environments use different branches in a source code repository. The Gitflow workflow is considered legacy, and the trunk-based workflow is the modern, preferred approach.

golden image

A snapshot of a system or software that is used as a template to deploy new instances of that system or software. For example, in manufacturing, a golden image can be used to provision software on multiple devices and helps improve speed, scalability, and productivity in device manufacturing operations.

greenfield strategy

The absence of existing infrastructure in a new environment. When adopting a greenfield strategy for a system architecture, you can select all new technologies without the restriction of compatibility with existing infrastructure, also known as brownfield. If you are expanding the existing infrastructure, you might blend brownfield and greenfield strategies.

guardrail

A high-level rule that helps govern resources, policies, and compliance across organizational units (OUs). *Preventive guardrails* enforce policies to ensure alignment to compliance standards. They are implemented by using service control policies and IAM permissions boundaries.

Detective guardrails detect policy violations and compliance issues, and generate alerts for remediation. They are implemented by using AWS Config, AWS Security Hub, Amazon GuardDuty, AWS Trusted Advisor, Amazon Inspector, and custom AWS Lambda checks.

H

HA

See high availability.

heterogeneous database migration

Migrating your source database to a target database that uses a different database engine (for example, Oracle to Amazon Aurora). Heterogeneous migration is typically part of a re-architecting effort, and converting the schema can be a complex task. AWS provides AWS SCT that helps with schema conversions.

high availability (HA)

The ability of a workload to operate continuously, without intervention, in the event of challenges or disasters. HA systems are designed to automatically fail over, consistently deliver high-quality performance, and handle different loads and failures with minimal performance impact.

historian modernization

An approach used to modernize and upgrade operational technology (OT) systems to better serve the needs of the manufacturing industry. A *historian* is a type of database that is used to collect and store data from various sources in a factory.

holdout data

A portion of historical, labeled data that is withheld from a dataset that is used to train a machine learning model. You can use holdout data to evaluate the model performance by comparing the model predictions against the holdout data.

homogeneous database migration

Migrating your source database to a target database that shares the same database engine (for example, Microsoft SQL Server to Amazon RDS for SQL Server). Homogeneous migration is typically part of a rehosting or replatforming effort. You can use native database utilities to migrate the schema.

hot data

Data that is frequently accessed, such as real-time data or recent translational data. This data typically requires a high-performance storage tier or class to provide fast query responses.

hotfix

An urgent fix for a critical issue in a production environment. Due to its urgency, a hotfix is usually made outside of the typical DevOps release workflow.

hypercare period

Immediately following cutover, the period of time when a migration team manages and monitors the migrated applications in the cloud in order to address any issues. Typically, this period is 1–4 days in length. At the end of the hypercare period, the migration team typically transfers responsibility for the applications to the cloud operations team.

I

IaC

See infrastructure as code.

identity-based policy

A policy attached to one or more IAM principals that defines their permissions within the AWS Cloud environment.

idle application

An application that has an average CPU and memory usage between 5 and 20 percent over a period of 90 days. In a migration project, it is common to retire these applications or retain them on premises.

IIoT

See industrial Internet of Things.

immutable infrastructure

A model that deploys new infrastructure for production workloads instead of updating, patching, or modifying the existing infrastructure. Immutable infrastructures are inherently more consistent, reliable, and predictable than mutable infrastructure. For more information, see the Deploy using immutable infrastructure best practice in the AWS Well-Architected Framework.

inbound (ingress) VPC

In an AWS multi-account architecture, a VPC that accepts, inspects, and routes network connections from outside an application. The AWS Security Reference Architecture recommends setting up your Network account with inbound, outbound, and inspection VPCs to protect the two-way interface between your application and the broader internet.

incremental migration

A cutover strategy in which you migrate your application in small parts instead of performing a single, full cutover. For example, you might move only a few microservices or users to the new system initially. After you verify that everything is working properly, you can incrementally move additional microservices or users until you can decommission your legacy system. This strategy reduces the risks associated with large migrations.

Industry 4.0

A term that was introduced by Klaus Schwab in 2016 to refer to the modernization of manufacturing processes through advances in connectivity, real-time data, automation, analytics, and AI/ML.

infrastructure

All of the resources and assets contained within an application's environment.

infrastructure as code (IaC)

The process of provisioning and managing an application's infrastructure through a set of configuration files. IaC is designed to help you centralize infrastructure management, standardize resources, and scale quickly so that new environments are repeatable, reliable, and consistent.

industrial Internet of Things (IIoT)

The use of internet-connected sensors and devices in the industrial sectors, such as manufacturing, energy, automotive, healthcare, life sciences, and agriculture. For more information, see Building an industrial Internet of Things (IIoT) digital transformation strategy.

inspection VPC

In an AWS multi-account architecture, a centralized VPC that manages inspections of network traffic between VPCs (in the same or different AWS Regions), the internet, and on-premises networks. The AWS Security Reference Architecture recommends setting up your Network account with inbound, outbound, and inspection VPCs to protect the two-way interface between your application and the broader internet.

Internet of Things (IoT)

The network of connected physical objects with embedded sensors or processors that communicate with other devices and systems through the internet or over a local communication network. For more information, see What is IoT?

interpretability

A characteristic of a machine learning model that describes the degree to which a human can understand how the model's predictions depend on its inputs. For more information, see Machine learning model interpretability with AWS.

IoT

See Internet of Things.

IT information library (ITIL)

A set of best practices for delivering IT services and aligning these services with business requirements. ITIL provides the foundation for ITSM.

IT service management (ITSM)

Activities associated with designing, implementing, managing, and supporting IT services for an organization. For information about integrating cloud operations with ITSM tools, see the operations integration guide.

ITIL

See IT information library.

ITSM

See IT service management.

L

label-based access control (LBAC)

An implementation of mandatory access control (MAC) where the users and the data itself are each explicitly assigned a security label value. The intersection between the user security label and data security label determines which rows and columns can be seen by the user.

landing zone

A landing zone is a well-architected, multi-account AWS environment that is scalable and secure. This is a starting point from which your organizations can quickly launch and deploy workloads and applications with confidence in their security and infrastructure environment. For more information about landing zones, see Setting up a secure and scalable multi-account AWS environment.

large language model (LLM)

A deep learning AI model that is pretrained on a vast amount of data. An LLM can perform multiple tasks, such as answering questions, summarizing documents, translating text into other languages, and completing sentences. For more information, see What are LLMs.

large migration

A migration of 300 or more servers.

LBAC

See label-based access control.

least privilege

The security best practice of granting the minimum permissions required to perform a task. For more information, see Apply least-privilege permissions in the IAM documentation.

lift and shift

See 7 Rs.

little-endian system

A system that stores the least significant byte first. See also endianness.

LLM

See large language model.

lower environments

See environment.

M

machine learning (ML)

A type of artificial intelligence that uses algorithms and techniques for pattern recognition and learning. ML analyzes and learns from recorded data, such as Internet of Things (IoT) data, to generate a statistical model based on patterns. For more information, see Machine Learning.

main branch

See branch.

malware

Software that is designed to compromise computer security or privacy. Malware might disrupt computer systems, leak sensitive information, or gain unauthorized access. Examples of malware include viruses, worms, ransomware, Trojan horses, spyware, and keyloggers.

managed services

AWS services for which AWS operates the infrastructure layer, the operating system, and platforms, and you access the endpoints to store and retrieve data. Amazon Simple Storage Service (Amazon S3) and Amazon DynamoDB are examples of managed services. These are also known as *abstracted services*.

manufacturing execution system (MES)

A software system for tracking, monitoring, documenting, and controlling production processes that convert raw materials to finished products on the shop floor.

MAP

See Migration Acceleration Program.

mechanism

A complete process in which you create a tool, drive adoption of the tool, and then inspect the results in order to make adjustments. A mechanism is a cycle that reinforces and improves itself as it operates. For more information, see Building mechanisms in the AWS Well-Architected Framework.

member account

All AWS accounts other than the management account that are part of an organization in AWS Organizations. An account can be a member of only one organization at a time.

MES

See manufacturing execution system.

Message Queuing Telemetry Transport (MQTT)

A lightweight, machine-to-machine (M2M) communication protocol, based on the publish/subscribe pattern, for resource-constrained IoT devices.

microservice

A small, independent service that communicates over well-defined APIs and is typically owned by small, self-contained teams. For example, an insurance system might include microservices that map to business capabilities, such as sales or marketing, or subdomains, such as purchasing, claims, or analytics. The benefits of microservices include agility, flexible scaling, easy deployment, reusable code, and resilience. For more information, see Integrating microservices by using AWS serverless services.

microservices architecture

An approach to building an application with independent components that run each application process as a microservice. These microservices communicate through a well-defined interface by using lightweight APIs. Each microservice in this architecture can be updated, deployed, and scaled to meet demand for specific functions of an application. For more information, see Implementing microservices on AWS.

Migration Acceleration Program (MAP)

An AWS program that provides consulting support, training, and services to help organizations build a strong operational foundation for moving to the cloud, and to help offset the initial cost of migrations. MAP includes a migration methodology for executing legacy migrations in a methodical way and a set of tools to automate and accelerate common migration scenarios.

migration at scale

The process of moving the majority of the application portfolio to the cloud in waves, with more applications moved at a faster rate in each wave. This phase uses the best practices and lessons learned from the earlier phases to implement a *migration factory* of teams, tools, and processes to streamline the migration of workloads through automation and agile delivery. This is the third phase of the AWS migration strategy.

migration factory

Cross-functional teams that streamline the migration of workloads through automated, agile approaches. Migration factory teams typically include operations, business analysts and owners, migration engineers, developers, and DevOps professionals working in sprints. Between 20 and 50 percent of an enterprise application portfolio consists of repeated patterns that can be optimized by a factory approach. For more information, see the discussion of migration factories and the Cloud Migration Factory guide in this content set.

migration metadata

The information about the application and server that is needed to complete the migration. Each migration pattern requires a different set of migration metadata. Examples of migration metadata include the target subnet, security group, and AWS account.

migration pattern

A repeatable migration task that details the migration strategy, the migration destination, and the migration application or service used. Example: Rehost migration to Amazon EC2 with AWS Application Migration Service.

Migration Portfolio Assessment (MPA)

An online tool that provides information for validating the business case for migrating to the AWS Cloud. MPA provides detailed portfolio assessment (server right-sizing, pricing, TCO comparisons, migration cost analysis) as well as migration planning (application data analysis and data collection, application grouping, migration prioritization, and wave planning). The MPA tool (requires login) is available free of charge to all AWS consultants and APN Partner consultants.

Migration Readiness Assessment (MRA)

The process of gaining insights about an organization's cloud readiness status, identifying strengths and weaknesses, and building an action plan to close identified gaps, using the AWS CAF. For more information, see the migration readiness guide. MRA is the first phase of the AWS migration strategy.

migration strategy

The approach used to migrate a workload to the AWS Cloud. For more information, see the 7 Rs entry in this glossary and see Mobilize your organization to accelerate large-scale migrations.

ML

See machine learning.

modernization

Transforming an outdated (legacy or monolithic) application and its infrastructure into an agile, elastic, and highly available system in the cloud to reduce costs, gain efficiencies, and take advantage of innovations. For more information, see Strategy for modernizing applications in the AWS Cloud.

modernization readiness assessment

An evaluation that helps determine the modernization readiness of an organization's applications; identifies benefits, risks, and dependencies; and determines how well the organization can support the future state of those applications. The outcome of the assessment is a blueprint of the target architecture, a roadmap that details development phases and milestones for the modernization process, and an action plan for addressing identified gaps. For more information, see Evaluating modernization readiness for applications in the AWS Cloud.

monolithic applications (monoliths)

Applications that run as a single service with tightly coupled processes. Monolithic applications have several drawbacks. If one application feature experiences a spike in demand, the entire architecture must be scaled. Adding or improving a monolithic application's features also becomes more complex when the code base grows. To address these issues, you can use a microservices architecture. For more information, see Decomposing monoliths into microservices.

MPA

See Migration Portfolio Assessment.

MQTT

See Message Queuing Telemetry Transport.

multiclass classification

A process that helps generate predictions for multiple classes (predicting one of more than two outcomes). For example, an ML model might ask "Is this product a book, car, or phone?" or "Which product category is most interesting to this customer?"

mutable infrastructure

A model that updates and modifies the existing infrastructure for production workloads. For improved consistency, reliability, and predictability, the AWS Well-Architected Framework recommends the use of immutable infrastructure as a best practice.

O

OAC

See origin access control.

OAI

See origin access identity.

OCM

See organizational change management.

offline migration

A migration method in which the source workload is taken down during the migration process. This method involves extended downtime and is typically used for small, non-critical workloads.

OI

See operations integration.

OLA

See operational-level agreement.

online migration

A migration method in which the source workload is copied to the target system without being taken offline. Applications that are connected to the workload can continue to function during the migration. This method involves zero to minimal downtime and is typically used for critical production workloads.

OPC-UA

See Open Process Communications - Unified Architecture.

Open Process Communications - Unified Architecture (OPC-UA)

A machine-to-machine (M2M) communication protocol for industrial automation. OPC-UA provides an interoperability standard with data encryption, authentication, and authorization schemes.

operational-level agreement (OLA)

An agreement that clarifies what functional IT groups promise to deliver to each other, to support a service-level agreement (SLA).

operational readiness review (ORR)

A checklist of questions and associated best practices that help you understand, evaluate, prevent, or reduce the scope of incidents and possible failures. For more information, see Operational Readiness Reviews (ORR) in the AWS Well-Architected Framework.

operational technology (OT)

Hardware and software systems that work with the physical environment to control industrial operations, equipment, and infrastructure. In manufacturing, the integration of OT and information technology (IT) systems is a key focus for Industry 4.0 transformations.

operations integration (OI)

The process of modernizing operations in the cloud, which involves readiness planning, automation, and integration. For more information, see the operations integration guide.

organization trail

A trail that's created by AWS CloudTrail that logs all events for all AWS accounts in an organization in AWS Organizations. This trail is created in each AWS account that's part of the organization and tracks the activity in each account. For more information, see Creating a trail for an organization in the CloudTrail documentation.

organizational change management (OCM)

A framework for managing major, disruptive business transformations from a people, culture, and leadership perspective. OCM helps organizations prepare for, and transition to, new systems and strategies by accelerating change adoption, addressing transitional issues, and driving cultural and organizational changes. In the AWS migration strategy, this framework is called *people acceleration*, because of the speed of change required in cloud adoption projects. For more information, see the OCM guide.

origin access control (OAC)

In CloudFront, an enhanced option for restricting access to secure your Amazon Simple Storage Service (Amazon S3) content. OAC supports all S3 buckets in all AWS Regions, server-side encryption with AWS KMS (SSE-KMS), and dynamic `PUT` and `DELETE` requests to the S3 bucket.

origin access identity (OAI)

In CloudFront, an option for restricting access to secure your Amazon S3 content. When you use OAI, CloudFront creates a principal that Amazon S3 can authenticate with. Authenticated principals can access content in an S3 bucket only through a specific CloudFront distribution. See also OAC, which provides more granular and enhanced access control.

ORR

See operational readiness review.

OT

See operational technology.

outbound (egress) VPC

In an AWS multi-account architecture, a VPC that handles network connections that are initiated from within an application. The AWS Security Reference Architecture recommends setting up your Network account with inbound, outbound, and inspection VPCs to protect the two-way interface between your application and the broader internet.

P

permissions boundary

An IAM management policy that is attached to IAM principals to set the maximum permissions that the user or role can have. For more information, see Permissions boundaries in the IAM documentation.

personally identifiable information (PII)

Information that, when viewed directly or paired with other related data, can be used to reasonably infer the identity of an individual. Examples of PII include names, addresses, and contact information.

PII

See personally identifiable information.

playbook

A set of predefined steps that capture the work associated with migrations, such as delivering core operations functions in the cloud. A playbook can take the form of scripts, automated runbooks, or a summary of processes or steps required to operate your modernized environment.

PLC

See programmable logic controller.

PLM

See product lifecycle management.

policy

An object that can define permissions (see identity-based policy), specify access conditions (see resource-based policy), or define the maximum permissions for all accounts in an organization in AWS Organizations (see service control policy).

polyglot persistence

Independently choosing a microservice's data storage technology based on data access patterns and other requirements. If your microservices have the same data storage technology, they can encounter implementation challenges or experience poor performance. Microservices are more easily implemented and achieve better performance and scalability if they use the data store best adapted to their requirements. For more information, see Enabling data persistence in microservices.

portfolio assessment

A process of discovering, analyzing, and prioritizing the application portfolio in order to plan the migration. For more information, see Evaluating migration readiness.

predicate

A query condition that returns `true` or `false`, commonly located in a `WHERE` clause.

predicate pushdown

A database query optimization technique that filters the data in the query before transfer. This reduces the amount of data that must be retrieved and processed from the relational database, and it improves query performance.

preventative control

A security control that is designed to prevent an event from occurring. These controls are a first line of defense to help prevent unauthorized access or unwanted changes to your network. For more information, see Preventative controls in *Implementing security controls on AWS*.

principal

An entity in AWS that can perform actions and access resources. This entity is typically a root user for an AWS account, an IAM role, or a user. For more information, see *Principal* in Roles terms and concepts in the IAM documentation.

privacy by design

A system engineering approach that takes privacy into account through the whole development process.

private hosted zones

A container that holds information about how you want Amazon Route 53 to respond to DNS queries for a domain and its subdomains within one or more VPCs. For more information, see Working with private hosted zones in the Route 53 documentation.

proactive control

A security control designed to prevent the deployment of noncompliant resources. These controls scan resources before they are provisioned. If the resource is not compliant with the control, then it isn't provisioned. For more information, see the Controls reference guide in the AWS Control Tower documentation and see Proactive controls in *Implementing security controls on AWS*.

product lifecycle management (PLM)

The management of data and processes for a product throughout its entire lifecycle, from design, development, and launch, through growth and maturity, to decline and removal.

production environment

See environment.

programmable logic controller (PLC)

In manufacturing, a highly reliable, adaptable computer that monitors machines and automates manufacturing processes.

prompt chaining

Using the output of one LLM prompt as the input for the next prompt to generate better responses. This technique is used to break down a complex task into subtasks, or to iteratively refine or expand a preliminary response. It helps improve the accuracy and relevance of a model’s responses and allows for more granular, personalized results.

pseudonymization

The process of replacing personal identifiers in a dataset with placeholder values. Pseudonymization can help protect personal privacy. Pseudonymized data is still considered to be personal data.

publish/subscribe (pub/sub)

A pattern that enables asynchronous communications among microservices to improve scalability and responsiveness. For example, in a microservices-based MES, a microservice can publish event messages to a channel that other microservices can subscribe to. The system can add new microservices without changing the publishing service.

Q

query plan

A series of steps, like instructions, that are used to access the data in a SQL relational database system.

query plan regression

When a database service optimizer chooses a less optimal plan than it did before a given change to the database environment. This can be caused by changes to statistics, constraints, environment settings, query parameter bindings, and updates to the database engine.

R

RACI matrix

See responsible, accountable, consulted, informed (RACI).

RAG

See Retrieval Augmented Generation.

ransomware

A malicious software that is designed to block access to a computer system or data until a payment is made.

RASCI matrix

See responsible, accountable, consulted, informed (RACI).

RCAC

See row and column access control.

read replica

A copy of a database that’s used for read-only purposes. You can route queries to the read replica to reduce the load on your primary database.

re-architect

See 7 Rs.

recovery point objective (RPO)

The maximum acceptable amount of time since the last data recovery point. This determines what is considered an acceptable loss of data between the last recovery point and the interruption of service.

recovery time objective (RTO)

The maximum acceptable delay between the interruption of service and restoration of service.

refactor

See 7 Rs.

Region

A collection of AWS resources in a geographic area. Each AWS Region is isolated and independent of the others to provide fault tolerance, stability, and resilience. For more information, see Specify which AWS Regions your account can use.

regression

An ML technique that predicts a numeric value. For example, to solve the problem of "What price will this house sell for?" an ML model could use a linear regression model to predict a house's sale price based on known facts about the house (for example, the square footage).

rehost

See 7 Rs.

release

In a deployment process, the act of promoting changes to a production environment.

relocate

See 7 Rs.

replatform

See 7 Rs.

repurchase

See 7 Rs.

resiliency

An application's ability to resist or recover from disruptions. High availability and disaster recovery are common considerations when planning for resiliency in the AWS Cloud. For more information, see AWS Cloud Resilience.

resource-based policy

A policy attached to a resource, such as an Amazon S3 bucket, an endpoint, or an encryption key. This type of policy specifies which principals are allowed access, supported actions, and any other conditions that must be met.

responsible, accountable, consulted, informed (RACI) matrix

A matrix that defines the roles and responsibilities for all parties involved in migration activities and cloud operations. The matrix name is derived from the responsibility types defined in the matrix: responsible (R), accountable (A), consulted (C), and informed (I). The support (S) type is optional. If you include support, the matrix is called a *RASCI matrix*, and if you exclude it, it’s called a *RACI matrix*.

responsive control

A security control that is designed to drive remediation of adverse events or deviations from your security baseline. For more information, see Responsive controls in *Implementing security controls on AWS*.

retain

See 7 Rs.

retire

See 7 Rs.

Retrieval Augmented Generation (RAG)

A generative AI technology in which an LLM references an authoritative data source that is outside of its training data sources before generating a response. For example, a RAG model might perform a semantic search of an organization's knowledge base or custom data. For more information, see What is RAG.

rotation

The process of periodically updating a secret to make it more difficult for an attacker to access the credentials.

row and column access control (RCAC)

The use of basic, flexible SQL expressions that have defined access rules. RCAC consists of row permissions and column masks.

RPO

See recovery point objective.

RTO

See recovery time objective.

runbook

A set of manual or automated procedures required to perform a specific task. These are typically built to streamline repetitive operations or procedures with high error rates.

S

SAML 2.0

An open standard that many identity providers (IdPs) use. This feature enables federated single sign-on (SSO), so users can log into the AWS Management Console or call the AWS API operations without you having to create user in IAM for everyone in your organization. For more information about SAML 2.0-based federation, see About SAML 2.0-based federation in the IAM documentation.

SCADA

See supervisory control and data acquisition.

SCP

See service control policy.

secret

In AWS Secrets Manager, confidential or restricted information, such as a password or user credentials, that you store in encrypted form. It consists of the secret value and its metadata.

The secret value can be binary, a single string, or multiple strings. For more information, see What's in a Secrets Manager secret? in the Secrets Manager documentation.

security by design

A system engineering approach that takes security into account through the whole development process.

security control

A technical or administrative guardrail that prevents, detects, or reduces the ability of a threat actor to exploit a security vulnerability. There are four primary types of security controls: preventative, detective, responsive, and proactive.

security hardening

The process of reducing the attack surface to make it more resistant to attacks. This can include actions such as removing resources that are no longer needed, implementing the security best practice of granting least privilege, or deactivating unnecessary features in configuration files.

security information and event management (SIEM) system

Tools and services that combine security information management (SIM) and security event management (SEM) systems. A SIEM system collects, monitors, and analyzes data from servers, networks, devices, and other sources to detect threats and security breaches, and to generate alerts.

security response automation

A predefined and programmed action that is designed to automatically respond to or remediate a security event. These automations serve as detective or responsive security controls that help you implement AWS security best practices. Examples of automated response actions include modifying a VPC security group, patching an Amazon EC2 instance, or rotating credentials.

server-side encryption

Encryption of data at its destination, by the AWS service that receives it.

service control policy (SCP)

A policy that provides centralized control over permissions for all accounts in an organization in AWS Organizations. SCPs define guardrails or set limits on actions that an administrator can delegate to users or roles. You can use SCPs as allow lists or deny lists, to specify which services or actions are permitted or prohibited. For more information, see Service control policies in the AWS Organizations documentation.

service endpoint

The URL of the entry point for an AWS service. You can use the endpoint to connect programmatically to the target service. For more information, see AWS service endpoints in *AWS General Reference*.

service-level agreement (SLA)

An agreement that clarifies what an IT team promises to deliver to their customers, such as service uptime and performance.

service-level indicator (SLI)

A measurement of a performance aspect of a service, such as its error rate, availability, or throughput.

service-level objective (SLO)

A target metric that represents the health of a service, as measured by a service-level indicator.

shared responsibility model

A model describing the responsibility you share with AWS for cloud security and compliance. AWS is responsible for security *of* the cloud, whereas you are responsible for security *in* the cloud. For more information, see Shared responsibility model.

SIEM

See security information and event management system.

single point of failure (SPOF)

A failure in a single, critical component of an application that can disrupt the system.

SLA

See service-level agreement.

SLI

See service-level indicator.

SLO

See service-level objective.

split-and-seed model

A pattern for scaling and accelerating modernization projects. As new features and product releases are defined, the core team splits up to create new product teams. This helps scale your organization's capabilities and services, improves developer productivity, and supports rapid innovation. For more information, see Phased approach to modernizing applications in the AWS Cloud.

SPOF

See single point of failure.

star schema

A database organizational structure that uses one large fact table to store transactional or measured data and uses one or more smaller dimensional tables to store data attributes. This structure is designed for use in a data warehouse or for business intelligence purposes.

strangler fig pattern

An approach to modernizing monolithic systems by incrementally rewriting and replacing system functionality until the legacy system can be decommissioned. This pattern uses the analogy of a fig vine that grows into an established tree and eventually overcomes and replaces its host. The pattern was introduced by Martin Fowler as a way to manage risk when rewriting monolithic systems. For an example of how to apply this pattern, see Modernizing legacy Microsoft ASP.NET (ASMX) web services incrementally by using containers and Amazon API Gateway.

subnet

A range of IP addresses in your VPC. A subnet must reside in a single Availability Zone.

supervisory control and data acquisition (SCADA)

In manufacturing, a system that uses hardware and software to monitor physical assets and production operations.

symmetric encryption

An encryption algorithm that uses the same key to encrypt and decrypt the data.

synthetic testing

Testing a system in a way that simulates user interactions to detect potential issues or to monitor performance. You can use Amazon CloudWatch Synthetics to create these tests.

system prompt

A technique for providing context, instructions, or guidelines to an LLM to direct its behavior. System prompts help set context and establish rules for interactions with users.

T

tags

Key-value pairs that act as metadata for organizing your AWS resources. Tags can help you manage, identify, organize, search for, and filter resources. For more information, see Tagging your AWS resources.

target variable

The value that you are trying to predict in supervised ML. This is also referred to as an *outcome variable*. For example, in a manufacturing setting the target variable could be a product defect.

task list

A tool that is used to track progress through a runbook. A task list contains an overview of the runbook and a list of general tasks to be completed. For each general task, it includes the estimated amount of time required, the owner, and the progress.

test environment

See environment.

training

To provide data for your ML model to learn from. The training data must contain the correct answer. The learning algorithm finds patterns in the training data that map the input data attributes to the target (the answer that you want to predict). It outputs an ML model that captures these patterns. You can then use the ML model to make predictions on new data for which you don’t know the target.

transit gateway

A network transit hub that you can use to interconnect your VPCs and on-premises networks. For more information, see What is a transit gateway in the AWS Transit Gateway documentation.

trunk-based workflow

An approach in which developers build and test features locally in a feature branch and then merge those changes into the main branch. The main branch is then built to the development, preproduction, and production environments, sequentially.

trusted access

Granting permissions to a service that you specify to perform tasks in your organization in AWS Organizations and in its accounts on your behalf. The trusted service creates a service-linked role in each account, when that role is needed, to perform management tasks for you. For more information, see Using AWS Organizations with other AWS services in the AWS Organizations documentation.

tuning

To change aspects of your training process to improve the ML model's accuracy. For example, you can train the ML model by generating a labeling set, adding labels, and then repeating these steps several times under different settings to optimize the model.

two-pizza team

A small DevOps team that you can feed with two pizzas. A two-pizza team size ensures the best possible opportunity for collaboration in software development.

U

uncertainty

A concept that refers to imprecise, incomplete, or unknown information that can undermine the reliability of predictive ML models. There are two types of uncertainty: *Epistemic uncertainty* is caused by limited, incomplete data, whereas *aleatoric uncertainty* is caused by the noise and randomness inherent in the data. For more information, see the Quantifying uncertainty in deep learning systems guide.

undifferentiated tasks

Also known as *heavy lifting*, work that is necessary to create and operate an application but that doesn’t provide direct value to the end user or provide competitive advantage. Examples of undifferentiated tasks include procurement, maintenance, and capacity planning.

upper environments

See environment.

V

vacuuming

A database maintenance operation that involves cleaning up after incremental updates to reclaim storage and improve performance.

version control

Processes and tools that track changes, such as changes to source code in a repository.

VPC peering

A connection between two VPCs that allows you to route traffic by using private IP addresses. For more information, see What is VPC peering in the Amazon VPC documentation.

vulnerability

A software or hardware flaw that compromises the security of the system.

W

warm cache

A buffer cache that contains current, relevant data that is frequently accessed. The database instance can read from the buffer cache, which is faster than reading from the main memory or disk.

warm data

Data that is infrequently accessed. When querying this kind of data, moderately slow queries are typically acceptable.

window function

A SQL function that performs a calculation on a group of rows that relate in some way to the current record. Window functions are useful for processing tasks, such as calculating a moving average or accessing the value of rows based on the relative position of the current row.

workload

A collection of resources and code that delivers business value, such as a customer-facing application or backend process.

workstream

Functional groups in a migration project that are responsible for a specific set of tasks. Each workstream is independent but supports the other workstreams in the project. For example, the portfolio workstream is responsible for prioritizing applications, wave planning, and collecting migration metadata. The portfolio workstream delivers these assets to the migration workstream, which then migrates the servers and applications.

WORM

See write once, read many.

WQF

See AWS Workload Qualification Framework.

write once, read many (WORM)

A storage model that writes data a single time and prevents the data from being deleted or modified. Authorized users can read the data as many times as needed, but they cannot change it. This data storage infrastructure is considered immutable.

Z

zero-day exploit

An attack, typically malware, that takes advantage of a zero-day vulnerability.

zero-day vulnerability

An unmitigated flaw or vulnerability in a production system. Threat actors can use this type of vulnerability to attack the system. Developers frequently become aware of the vulnerability as a result of the attack.

zero-shot prompting

Providing an LLM with instructions for performing a task but no examples (*shots*) that can help guide it. The LLM must use its pre-trained knowledge to handle the task. The effectiveness of zero-shot prompting depends on the complexity of the task and the quality of the prompt. See also few-shot prompting.

zombie application

An application that has an average CPU and memory usage below 5 percent. In a migration project, it is common to retire these applications.